Notice of Allowability	Application No.	Applicant(s)	
	09/608,282	DODD ET AL.	
	Examiner	Art Unit	
	Tamara Teslovich	2137	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *Applicant's "Amendments and Response" filed 30 June 2004.*.

2. ☒ The allowed claim(s) is/are *2-12,15 and 17-52*.

3. ☒ The drawings filed on *30 June 2000* are accepted by the Examiner.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date 04.28.03 05.14.03
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date Attached (AC)
7. ☒ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

Andrew Caldwell

ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER

DETAILED ACTION

This action is in response to the Amendment filed on June 30, 2004.

Claims 1, 13-14, and 16 have been cancelled by applicant. Claims 2-12, 15 and 25 have been amended, and are herein considered. Claims 17-52 have been added and are herein considered.

Response to Arguments

Applicant's arguments filed June 30, 2004 have been fully considered and treated as follows:

Applicant's arguments, see pages 22-27, filed June 30, 2004 with respect to Claim 2 have been fully considered and are persuasive. The 35 U.S.C. § 103(a) Rejection of Claim 2 has been withdrawn.

Applicant's arguments, see page 27, filed June 30, 2004 with respect to Claims 3-5, 9-10, 12, and 15 have been fully considered and are persuasive. The 35 U.S.C. § 103(a) Rejection of Claims 3-5, 9-10, 12 and 15 has been withdrawn.

Applicant's arguments, see pages 27-28, filed June 30, 2004 with respect to Claims 17-52 have been fully considered and are persuasive. The new claims comprise allowable subject matter of allowable claims, finding clear support in the specifications and containing no new matter.

Examiner's Amendment

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with James M. Hannon on February 3, 2005.

The application has been amended as follows:

Please amend Claims in accordance with Examiner's "Amendment to Claims" included as pages 4-22 of this office action.

Please amend Abstract in accordance with Examiner's "Amendment to the Abstract" included as page 23 of this office action.

AMENDMENT TO CLAIMS

1. (Canceled)

2. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising an engine, exploit manager, resource manager, and built-in exploits, comprising the steps of:

updating a capability of the scanner to conduct vulnerability assessments of the host computer system by obtaining a pluggable express update package, wherein the update package is configured as an independent plug-in module that is separate from the scanner and communicates with the scanner to support the vulnerability assessments by the scanner, the update package comprising:

an exploit plug-in module comprising exploit objects for exploits that check the host computer system for at least certain ones of the security vulnerabilities, the exploits representing modifications or updates to the built-in exploits of the scanner;

a resource plug-in module comprising resource objects representing resources that can be used by the scanner, the resources maintained as resource objects separate from the exploits of the exploit objects to support an independent updating of the resource objects and the exploit objects;

a dat file comprising exploit attribute information defining attribute information for the exploits of the exploit plug-in module, the exploit attribute information stored in a file separate from the exploit objects to support an independent updating of the dat file and the exploit objects; and

a help file comprising on-line help information about the exploits of the exploit plug-in module, the help information stored in a file separate from the exploit objects to support an independent updating of the help file and the exploit objects;

supplying the exploit attribute information to the exploit manager from the dat file;

passing the exploit objects and the resource objects from the exploit manager and the resource manager to an engine of the scanner; and

executing the exploits of the exploit plug-in module at the scanner.

3. (Previously Presented) The computer-implemented process of claim 2 wherein each of said resources can be assigned a namespace based upon the resource's scope.

4. (Previously Presented) The computer-implemented process of claim 2, wherein the step of executing exploits comprises the steps of:

running standard built-in exploits of the scanner;

running standard plug-in exploits of the pluggable express update package;

running denial of service plug-in exploits of the pluggable express update package; and

running denial of service built-in exploits of the scanner.

5. (Previously Presented) The computer-implemented process of claim 4, wherein said steps of running standard and denial of service built-in exploits of the scanner comprises ~~includes~~ the steps of:

retrieving one of the built-in exploits at the top of a run-order list maintained by the scanner;

running the retrieved exploit;

recording exploit result information to a database and a scanner log file;

sending the exploit result information to a user interface; and

repeating the above steps for the remaining built-in exploits.

6. (Previously Presented) The computer-implemented process of claim 4, wherein the steps of running standard and denial of service plug-in exploits of the pluggable express update package comprises the steps of:

copying from a session object a master exploit list and a master resource list;

obtaining exploit information from a scanpolicy object for one of the plug-in exploits;

creating a target object and placing the exploit information in the target object;

passing the target object to one of the exploit objects associated with the plug-in exploit;

running the plug-in exploit;

adding exploit result information to the target object;

passing the target object back to a plug-in engine of the scanner;

querying the target object for the exploit result information;

recording the exploit result information to a scanner log file and sending the exploit result information to a user interface; and

repeating the above steps for the remaining plug-in exploits.

7. (Previously Presented) The computer-implemented process of claim 6, wherein said step of repeating the above steps for the remaining plug-in exploits comprises the steps of:

running a plug-in exploit that neither produces nor consumes shared resources;

running a plug-in exploit that only produces at least one of the shared resources;

running a plug-in exploit that produces and consumes at least one of the shared resources; and

running a plug-in exploit that only consumes at least one of the shared resources.

8. (Previously Presented) The computer-implemented process of claim 7, wherein said step of running a plug-in exploit that produces and consumes at least one of the shared resources further comprises the step of ensuring that plug-in exploits that produce at least one of the shared resources consumed by the exploit are run before the plug-in exploit that produces and consumes at least one of the shared resources is run.

9. (Previously Presented) The computer-implemented process of claim 2, further comprising the step of initializing the scanner.

10. (Previously Presented) The computer-implemented process of 9, wherein the step of initializing a scanner comprises the steps of:

enumerating the exploit plug-in module and the resource plug-in module;

enumerating the exploit objects and the resource objects;

running a load security procedure for the exploit and the resource plug-in modules; and

initializing a policy manager comprising at least one security policy that is retrievable by the engine of the scanner.

11. (Previously Presented) The computer-implemented process of claim 10, wherein the step of initializing a policy manager comprises the steps of:

identifying available exploits and available resources;

identifying available exploit objects and available resource objects corresponding to the available exploits and available resources;

generating maps that identify the exploit and the resource plug-in modules containing the available exploit objects and the available resource objects;

creating the available exploit objects and the common-setting resource objects; and

querying the available exploit objects and the common-setting resource objects.

12. (Previously Presented) The computer-implemented process of claim 2, further comprising the step of receiving from a user interface a list of host computer systems the scan engine is authorized to scan, a list of the exploits for execution by the scanner, and the identity of at least one host computer system to scan for security vulnerabilities.

13-14. (Canceled)

15. (Previously Presented) The computer-implemented process of claim 2, comprising the steps of:

querying a session manager for available hosts to scan;

querying session objects for one of the available hosts; and

returning one of the available hosts to a host-scanning thread.

16. (Canceled)

17. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising an engine, an exploit manager, a resource manager, standard built-in exploits and denial of service built-in exploits, comprising the steps of:

installing an express update package comprising an exploit plug-in module having exploit objects representing exploits that check the host computer system for vulnerabilities, the exploits comprising standard plug-in exploits and denial of service plug-in exploits; a resource plug-in module having resource objects representing resources for use by the scanner; a dat file comprising exploit attribute information; and a help file comprising on-line help information;

supplying the exploit attribute information from the dat file to the exploit manager of the scanner;

passing information about the exploit objects and resource objects from the exploit manager and the resource manager to the scanner engine;

running the standard built-in exploits and the denial of service built-in exploits by the scanner engine;

running the standard plug-in exploits and the denial of service plug-in exploits by a plug-in engine of the scanner, wherein the step of running the standard plug-in exploits and the denial of service plug-in exploits comprises the steps of:

(a) obtaining copies of a master exploit list and a master resource list from a session object;

(b) obtaining exploit information from a scanpolicy object for an identified one of the plug-in exploits;

(c) creating a target object and placing the exploit information in the target object;

(d) passing the target object to one of the exploit objects corresponding to the identified plug-in exploit;

(e) running the identified plug-in exploit;

(f) adding exploit result information to the target object;

(g) passing the target object to the plug-in engine;

(h) querying the target object for the exploit result information;

(i) recording the exploit result information to a scanner log file and sending the exploit result information to a user interface; and

repeating steps (b) - (i) for each of the remaining standard and denial of service plug-in exploits.

18. (Currently Amended) The computer-implemented process of claim 17, wherein repeating the above steps for the remaining standard and denial of service plug-in exploits comprises the steps of:

running standard and denial of service plug-in exploits that neither produce nor consume at least one of the shared resources;

running standard and denial of service plug-in exploits that only produce at least one of the shared resources;

running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources; and

running standard and denial of service plug-in exploits that only consume at least one of the shared resources.

19. (Currently Amended) The computer-implemented process of claim 18, wherein said step of running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources further comprises the step of ensuring that standard and denial of service plug-in exploits that produce at least one shared resource consumed by a particular exploit are run before the particular exploit is run.

20. (Previously Presented) The computer-implemented process of claim 17 further comprising the steps of:

enumerating the exploit plug-in module and the resource plug-in module and the exploit and the resource objects;

running load security for each of the exploit and resource plug-in modules; and

initializing a policy manager comprising at least one security policy that is retrievable by the engine of the scanner.

21. (Previously Presented) The computer-implemented process of claim 20, wherein initializing a policy manager comprises the steps of:

identifying available exploits and available resources;

identifying available exploit objects and available resource objects corresponding to the available exploits and available resources; and

generating maps that identify the exploit plug-in module and the resource plug-in module containing the available exploit objects and the available resource objects.

22. (Previously Presented) The computer-implemented process of claim 17, further comprising the step of receiving from the user interface a list of host computer systems that the scanner is authorized to scan, a list of exploits to be used to check the host computer system for security vulnerabilities, and the identity of the host computer system.

23. (Previously Presented) The computer-implemented process of claim 17, comprising the steps of:

querying a session manager for an identity of at least one host computer system to scan; and

sending the identity of the at least one host computer system to the scanner engine.

[THIS AREA INTENTIONALLY LEFT BLANK]

24. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising a policy manager, an engine, an exploit manager and a resource manager, comprising the steps of:

installing an express update package comprising an exploit plug-in module having exploit objects representing exploits that check the host computer system for vulnerabilities, the exploits comprising standard plug-in exploits and denial of service plug-in exploits; a resource plug-in module having resource objects representing resources for use by the scanner; a dat file comprising exploit attribute information; and a help file comprising on-line help information;

initializing the scanner by completing the following steps:

enumerating the exploit plug-in module and the resource plug-in module and the exploit and the resource objects;

running load security for each of the exploit and resource plug-in modules; and

initializing the policy manager, wherein the step of initializing the policy manager comprises the steps of:

requesting the exploit manager and the resource manager to identify available ones of the exploits and the resources;

using the exploit manager and the resource manager to query a registry for available ones of the exploit objects and the resource objects;

creating maps by the exploit manager and the resource manager, the maps identifying the exploit and resource plug-in modules containing the available exploit objects and the available resource objects;

issuing a request to the exploit manager and the resource manager to request the available exploit objects and common-setting resource objects;

returning the available exploit objects and the common-setting resource objects to the policy manager; and

issuing a query from the policy manager to query the available exploit objects and the common-setting resource objects for corresponding exploit attribute information and resource configuration information;

supplying the exploit attribute information to the exploit manager from the dat file;

passing exploit object and resource object information from the exploit manager and the resource manager to the scanner engine; and

executing the exploits at the scanner engine.

25. (Previously Presented) The computer-implemented process of claim 24, further comprising the step of receiving from the user interface a request to scan at least one host computer system for security vulnerabilities, the request comprising:

a list of host computer systems that the scanner is authorized to scan;

a list of exploits to be used to check the host computer system for security vulnerabilities; and

the identity of at least one host computer system to scan for security vulnerabilities.

26. (Previously Presented) The computer-implemented process of claim 24, wherein the scanner further comprises built-in exploits comprising standard built-in exploits and denial of service built-in exploits.

27. (Previously Presented) The computer-implemented process of claim 26, wherein the step of executing exploits at the scanner engine comprises the steps of:

running the standard built-in exploits of the scanner;

running the standard plug-in exploits of the express update package;

running the denial of service plug-in exploits of the express update package; and

running the denial of service built-in exploits of the scanner.

28. (Previously Presented) The computer-implemented process of claim 27, wherein the steps of running the standard and denial of service built-in exploits of the scanner comprise the steps of:

retrieving one of the built-in exploits at the top of a run-order list maintained by the scanner;

running the retrieved built-in exploit;

recording exploit result information to a database and a log file of the scanner;

sending the exploit result information to a user interface of the scanner; and

repeating the above steps for the remaining built-in exploits.

29. (Previously Presented) The computer-implemented process of claim 27, wherein the steps of running the standard and denial of service plug-in exploits comprises the steps of:

creating a target object and placing the exploit attribute information in the target object;

passing the target object to one of the exploit objects;

running one of the plug-in exploits;

receiving exploit result information at the target object in response to running the plug-in exploit;

passing the target object back to the engine of the scanner;

recording the exploit result information to a log file of the scanner and passing the exploit result information to a user interface of the scanner; and

repeating the above steps for the remaining plug-in exploits.

30. (Currently Amended) The computer-implemented process of claim 29, wherein repeating the above steps for the remaining plug-in exploits comprises the steps of:

running a plug-in exploit that neither produces nor consumes shared resources;

running a plug-in exploit that only produces at least one of the shared resources;

running a plug-in exploit that produces and consumes at least one of the shared resources; and

running a plug-in exploit that only consumes at least one of the shared resources.

31. (Currently Amended) The computer-implemented process of claim 30, wherein the step of running a plug-in exploit that produces and consumes at least one of the shared resources further comprises the step of ensuring that the plug-in exploits that produce at least one shared resource consumed by the plug-in exploit are run before the plug-in exploit that produces and consumes at least one shared resource is run.

32. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising an engine, an exploit manager, a resource manager, standard built-in exploits and denial of service built-in exploits, and a user interface, comprising the steps of:

updating a capability of the scanner to conduct security vulnerability assessments of the host computer system by obtaining an update comprising an exploit plug-in module having exploit objects representing exploits that check the host computer system for vulnerabilities, the exploits comprising standard plug-in exploits and denial of service plug-in exploits; a resource plug-in module having resource objects representing resources for use by the scanner; and a file comprising exploit attribute information;

installing the update as an independent plug-in for operation in connection with the scanner;

supplying the exploit attribute information from the update to the exploit manager of the scanner;

passing information about the exploit objects and resource objects from the exploit manager and the resource manager to the scanner engine;

running the standard built-in exploits and the denial of service built-in exploits at the scanner engine;

running the standard plug-in exploits and the denial of service plug-in exploits at a plug-in engine of the scanner, wherein the step of running the standard plug-in exploits and the denial of service plug-in exploits comprises the steps of:

(a) obtaining copies of a master exploit list and a master resource list;

(b) obtaining host information and selected ones of the resources for an identified one of the plug-in exploits;

(c) providing the host information and the selected resources via a target object to one of the exploit objects corresponding to the identified plug-in exploit

(e) running the identified plug-in exploit at the plug-in engine;

(f) adding scan result information to the target object in response to running the identified plug-in exploit;

(g) obtaining the scan result information from the target object for presentation via the user interface of the scanner; and

repeating steps (b) - (g) for each of the remaining standard and denial of service plug-in exploits.

33. (Currently Amended) The computer-implemented process of claim 32, wherein repeating steps (b) – (g) for each of the remaining standard and denial of service plug-in exploits comprises the steps of:

running standard and denial of service plug-in exploits that neither produce nor consume at least one of the shared resources;

running standard and denial of service plug-in exploits that only produce at least one of the shared resources;

running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources; and

running standard and denial of service plug-in exploits that only consume at least one of the shared resources.

34. (Currently Amended) The computer-implemented process of claim 33, wherein said step of running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources further comprises the step of ensuring that standard and denial of service plug-in exploits that produce at least one shared resource consumed by a particular plug-in exploit are run before the particular plug-in exploit is run.

35. (Previously Presented) The computer-implemented process of claim 32, further comprising the step of receiving from the user interface:

a list of host computer systems that the scanner is authorized to scan;

a list of exploits to be used to check the host computer system for security vulnerabilities, wherein the list comprises a selection of built-in and plug-in exploits, said selection made from the built-in exploits and the master exploit list; and

the identity of at least one host computer system to scan for security vulnerabilities.

36. (Previously Presented) The computer-implemented process of claim 32, comprising the steps of:

querying a session manager for an identity of at least one host computer system to scan; and

sending the identity of the at least one host computer system to the scanner engine.

37. (Previously Presented) The computer-implemented process of claim 32 further comprising the steps of:

enumerating the exploit plug-in module and the resource plug-in module and the exploit and the resource objects;

running load security for each of the exploit and resource plug-in modules; and

initializing a policy manager comprising at least one security policy that is retrievable by the scanner engine.

38. (Previously Presented) The computer-implemented process of claim 37, wherein initializing a policy manager comprises the steps of:

identifying available exploits and available resources;

identifying available exploit objects and available resource objects corresponding to the available exploits and available resources; and

generating maps that identify the exploit plug-in module and the resource plug-in module containing the available exploit objects and the available resource objects.

[THIS AREA INTENTIONALLY LEFT BLANK]

39. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising an engine, an exploit manager, a resource manager, standard built-in exploits and denial of service built-in exploits, comprising the steps of:

updating a capability of the scanner to conduct security vulnerability assessments of the host computer system by obtaining an update comprising an exploit plug-in module having exploit objects representing exploits that check the host computer system for vulnerabilities, the exploits comprising standard plug-in exploits and denial of service plug-in exploits; a resource plug-in module having resource objects representing resources for use by the scanner; and a file comprising exploit attribute information;

installing the update as an independent plug-in for operation in connection with the scanner;

supplying the exploit attribute information from the update to the exploit manager of the scanner;

passing information about the exploit objects and resource objects from the exploit manager and the resource manager to the scanner engine;

running the standard built-in exploits and the denial of service built-in exploits at the scanner engine;

running the standard plug-in exploits and the denial of service plug-in exploits at a plug-in engine of the scanner, wherein the step of running the standard plug-in exploits and the denial of service plug-in exploits comprises the steps of:

(a) obtaining copies of a master exploit list and a master resource list;

(b) obtaining host information and selected ones of the resources for an identified one of the plug-in exploits;

(c) providing the host information and the selected resources via a target object to one of the exploit objects corresponding to the identified plug-in exploit

(e) running the identified plug-in exploit at the plug-in engine;

(f) adding scan result information to the target object in response to running the identified plug-in exploit;

(g) obtaining the scan result information from the target object for storage in a scanner log file; and

repeating steps (b) - (g) for each of the remaining standard and denial of service plug-in exploits.

40. (Currently Amended) The computer-implemented process of claim 39, wherein repeating steps (b) – (g) for each of the remaining standard and denial of service plug-in exploits comprises the steps of:

running standard and denial of service plug-in exploits that neither produce nor consume at least one of the shared resources;

running standard and denial of service plug-in exploits that only produce at least one of the shared resources;

running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources; and

running standard and denial of service plug-in exploits that only consume at least one of the shared resources.

41. (Currently Amended) The computer-implemented process of claim 40, wherein said step of running standard and denial of service plug-in exploits that produce and consume at least one of the shared resources further comprises the step of ensuring that standard and denial of service plug-in exploits that produce at least one shared resource consumed by a particular plug-in exploit are run before the particular plug-in exploit is run.

42. (Previously Presented) The computer-implemented process of claim 39, further comprising the step of receiving from a user interface:

a list of host computer systems that the scanner is authorized to scan;

a list of exploits to be used to check the host computer system for security vulnerabilities, wherein the list comprises a selection of built-in and plug-in exploits, said selection made from the built-in exploits of the scanner and the master exploit list; and

the identity of at least one host computer system to scan for security vulnerabilities.

43. (Previously Presented) The computer-implemented process of claim 39, further comprising the steps of:

querying a session manager for an identity of at least one host computer system to scan; and

sending the identity of the at least one host computer system to the scanner engine.

44. (Previously Presented) The computer-implemented process of claim 39, further comprising the steps of:

enumerating the exploit plug-in module and the resource plug-in module and the exploit and the resource objects;

running load security for each of the exploit and resource plug-in modules; and

initializing a policy manager comprising at least one security policy that is retrievable by the scanner engine.

45. (Previously Presented) The computer-implemented process of claim 44, wherein initializing a policy manager comprises the steps of:

identifying available exploits and available resources;

identifying available exploit objects and available resource objects corresponding to the available exploits and available resources; and

generating maps that identify the exploit plug-in module and the resource plug-in module containing the available exploit objects and the available resource objects.

[THIS AREA INTENTIONALLY LEFT BLANK]

46. (Previously Presented) A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner comprising a policy manager, an engine, an exploit manager and a resource manager, comprising the steps of:

updating a capability of the scanner to conduct security vulnerability assessments of the host computer system by obtaining an update comprising an exploit plug-in module having exploit objects representing exploits that check the host computer system for vulnerabilities, the exploits comprising standard plug-in exploits and denial of service plug-in exploits; a resource plug-in module having resource objects representing resources for use by the scanner; a dat file comprising exploit attribute information; and a help file comprising on-line help information;

installing the update for use by the scanner;

initializing the scanner by completing the following steps:

enumerating the exploit plug-in module and the resource plug-in module and the exploit and the resource objects;

running load security for each of the exploit and resource plug-in modules; and

initializing the policy manager, wherein the step of initializing the policy manager comprises the steps of:

identifying available ones of the exploits and the resources;

identifying the exploit and resource plug-in modules containing the available ones of the exploit objects and the resource objects corresponding to the available exploits and resources;

obtaining the available exploit objects and common-setting resource objects; and

querying the available exploit objects and the common-setting resource objects for corresponding exploit attribute information and resource configuration information;

supplying the exploit attribute information to the exploit manager from the update

passing exploit object and resource object information from the exploit manager and the resource manager to the scanner engine; and

executing the exploits at the scanner engine.

47. (Previously Presented) The computer-implemented process of claim 46, further comprising the step of receiving from a user interface a request to scan the host computer system for security vulnerabilities, the request comprising:

a list of host computer systems that the scanner is authorized to scan,;

a list of exploits to be used to check the host computer system for security vulnerabilities, the list comprising exploits selected from the available ones of the exploits; and

the identity of the host computer system to scan for security vulnerabilities.

48. (Previously Presented) The computer-implemented process of claim 46, wherein the scanner further comprises built-in exploits comprising standard built-in exploits and denial of service built-in exploits.

49. (Previously Presented) The computer-implemented process of claim 48, wherein the step of executing exploits at the scanner engine comprises the steps of:

running the standard built-in exploits of the scanner;

running the standard plug-in exploits of the update;

running the denial of service plug-in exploits of the update; and

running the denial of service built-in exploits of the scanner.

50. (Previously Presented) The computer-implemented process of claim 49, wherein the steps of running the standard and denial of service built-in exploits of the scanner comprise the steps of:

retrieving one of the built-in exploits from a list of built-in exploits maintained by the scanner;

running the retrieved built-in exploit against the host computer system;

recording exploit result information to a database of the scanner;

sending the exploit result information to a user interface of the scanner; and

repeating the above steps for the remaining built-in exploits.

51. (Previously Presented) The computer-implemented process of claim 46, wherein executing the exploits at the scanner engine comprises the steps of:

creating a target object and placing the exploit attribute information in the target object;

passing the target object to one of the exploit objects;

running one of the plug-in exploits;

receiving exploit result information at the target object in response to running one of the plug-in exploits;

passing the target object back to the scanner engine;

recording the exploit result information to a log file of the scanner and passing the exploit result information to a user interface of the scanner; and

repeating the above steps for the remaining plug-in exploits.

52. (Currently Amended) The computer-implemented process of claim 51, wherein repeating the above steps for the remaining plug-in exploits comprises the steps of:

running a plug-in exploit that neither produces nor consumes shared resources;

running a plug-in exploit that only produces at least one of the shared resources;

ensuring that the plug-in exploits that produce at least one of the shared resources consumed by the plug-in exploit are run before the plug-in exploit that produces and consumes at least one of the shared resources is run;

running a plug-in exploit that produces and consumes at least one of the shared resources; and

running a plug-in exploit that only consumes at least one of the shared resources.

AMENDMENT TO THE ABSTRACT

A method and system identifies, fixes, and updates security vulnerabilities in a host computer or host computers. The present invention can communicate between a scanner with plug-in capability, an operating system, and an express update package.———The architectural set-up can allow exploits within the scanner and exploits in the express update package to function with no knowledge of each other. The user also needs no knowledge of whether the exploits are within the scanner or the express update package. Mutual authentication procedures can enable the scanner to load only legitimate express update packages, and can provide that express update packages can only be loaded into legitimate scanners.

Conclusion

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Examiner's Response to Arguments."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tamara Teslovich whose telephone number is (571) 272-4241. The examiner can normally be reached on Mon-Fri 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571) 272-3868. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tamara Teslovich

February 3, 2005

Andrew Caldwell

ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER